What is Claimed:

1. A storage platform comprising:

a data store in which data stored therein is defined in terms of items, elements, and relationships, wherein an item is a unit of data storable in the data store and comprises one or more elements, an element is an instance of a type comprising one or more fields, and a relationship is a link between at least two items;

a set of schemas that define different types of items, elements, and relationships; and

an application programming interface comprising a class for each of the different items, elements, and relationships defined in the set of schemas.

2. The storage platform recited in claim 1, wherein data may also be stored in the data store in the form of an extension to an existing item type, and wherein the application programming interface comprises a class for each different item extension.

3. The storage platform recited in claim 1, wherein the class for each type of item, element, and relationship is generated automatically based on the set of schemas that define each type of item, element, and relationship.

4. The storage platform recited in claim 1, wherein the classes for each type of item, element, and relationship define a set of data classes, and wherein the application programming interface further comprises a second set of classes that define a common set of behaviors for the data classes.

5. The storage platform recited in claim 4, wherein the second set of classes comprise a first class that represents a storage platform scope and that provides the context for queries on the data store and a second class the represents the results of a query on the data store.

6. The storage platform recited in claim 1, further comprising a database engine on which the data store is implemented, and wherein the different types of items, elements, and relationships in the data store are implemented in the database engine as user-defined types (UDT).

7. The storage platform recited in claim 6, wherein the application programming interface provides a query model that enables application programmers to form queries based on various properties of the items in the data store, in a manner that insulates the application programmer from the details of the query language of the database engine.

8. A method for providing an application programming interface between an application program and a storage platform for storing, organizing, sharing, and searching data, wherein the storage platform comprises a data store in which data stored therein is defined in terms of items, elements, and relationships, wherein an item is a unit of data storable in the data store and comprises one or more elements, an element is an instance of a type comprising one or more fields, and a relationship is a link between at least two items, said method comprising the steps of:

providing a set of schemas that define different types of items, elements, and relationships; and

generating, as part of the application programming interface, a class for each of the different items, elements, and relationships defined in the set of schemas.

9. The method recited in claim 8, wherein data may also be stored in the data store in the form of an extension to an existing item type, and wherein the method further comprises generating a class for each different item extension.

10. The method recited in claim 9, wherein the generated classes for each type of item, element, and relationship define a set of data classes, and wherein the method further comprises

the step of providing, as an additional part of the application programming interface, a second set of classes that define a common set of behaviors for the data classes.

11. The method recited in claim 10, wherein the second set of classes comprise a first class that represents a storage platform scope and that provides the context for queries on the data store and a second class the represents the results of a query on the data store.

12. The method recited in claim 8, wherein the data store of the storage platform is implemented on a database engine, and wherein the method further comprises the step of implementing the different types of items, elements, and relationships in the data store as user-defined types (UDT) in the database engine.

13. An application programming interface between an application program and a storage platform for storing, organizing, sharing, and searching data, wherein the storage platform comprises a data store in which data stored therein is defined in terms of items, elements, and relationships, wherein an item is a unit of data storable in the data store and comprises one or more elements, an element is an instance of a type comprising one or more fields, and a relationship is a link between at least two items, and wherein a set of schemas define different types of items, elements, and relationships, the application programming interface comprising a class for each of the different items, elements, and relationships defined in the set of schemas.

14. The application programming interface recited in claim 13, wherein data may also be stored in the data store in the form of an extension to an existing item type, and wherein the application programming interface further comprises a class for each different item extension.

15. The application programming interface recited in claim 13, wherein the classes for each type of item, element, and relationship define a set of data classes, and wherein the application programming interface further comprises a second set of classes that define a common set of behaviors for the data classes.

16. The application programming interface recited in claim 15, wherein the second set of classes comprise a first class that represents a storage platform scope and that provides the context for queries on the data store and a second class the represents the results of a query on the data store.